Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2312597 A1 2002/01/05

(21) 2 312 597

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(22) Date de dépôt/Filing Date: 2000/07/05

(41) Mise à la disp. pub./Open to Public Insp.: 2002/01/05

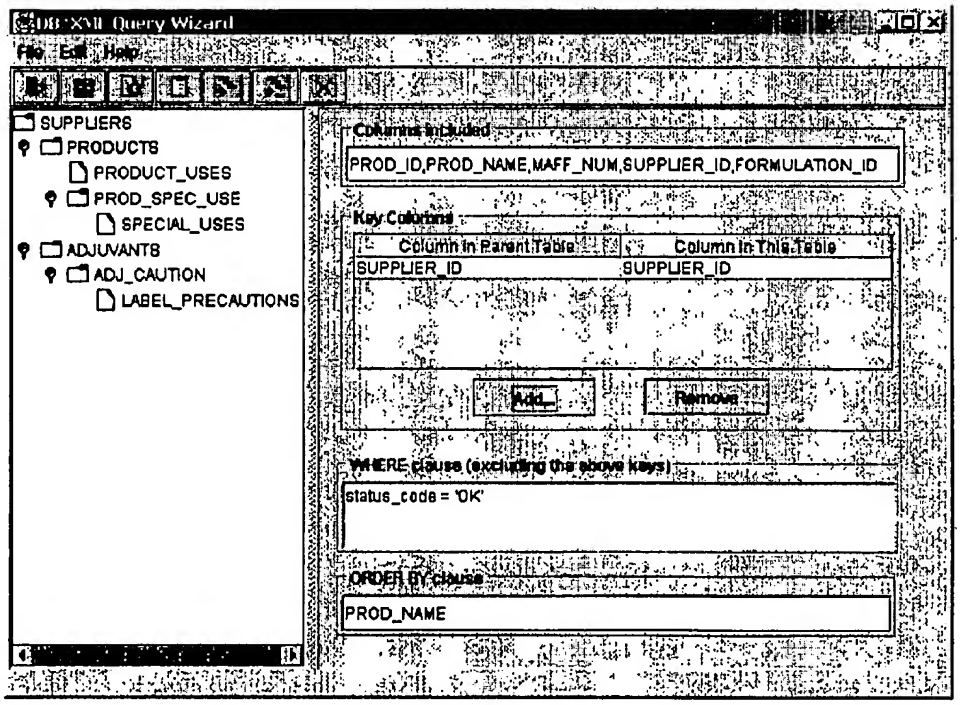
(51) Cl.Int.^{7}/Int.Cl.^{7} G06F 17/30, G06F 17/21

(71) Demandeur/Applicant:
ZHANG, BENJAMIN, CA

(72) Inventeur/Inventor:
ZHANG, BENJAMIN, CA

(54) Titre : METHODE DE RECUPERATION DE DONNEES UTILISANT LES INTERROGATIONS A STRUCTURE ARBORESCENTE ET RETOURNANT LES RESULTATS EN FORMAT XML

(54) Title: A METHOD FOR DATA RETRIEVAL USING TREE-STRUCTURED QUERY WITH RETURNED RESULT SET IN XML FORMAT



Sample of tree structure of a complex query and the properties for one of the nodes

(57) Abrégé/Abstract:
The present invention provides a technique by which complex queries can be defined and executed in a very flexible and efficient manner. It allows user to define the relationships between a parent and its different children, which can be nested to n-

Abstract

The present invention provides a technique by which complex queries can be defined and executed in a very flexible and efficient manner. It allows user to define the relationships between a parent and its different children, which can be nested to n-depth levels. The relationships are mapped to a special tree structure and the query processor executes the query based on the tree in an efficient way. The output data is also constructed in the defined tree structure in XML by default, which eliminates data redundancy. The output can be formatted either in Extensible Markup Language (XML) or HyperText Markup Language (HTML) format.

The present invention also provides two mechanisms to allow user to define the query: either through configuration files or through a graphical user interface. It is designed in such a way that it can be easily implemented as stand-alone application, for batch processing, or interacting with other applications. The query processing module and the graphical user interface modules are written in the Java programming language and the Java Database Connectivity (JDBC) technologies.

The technique of data retrieval disclosed in this invention is different from existing techniques in its high degree of flexibility and complexity in terms of the query structure, yet efficient processing and accurate output result. Because the output is also in tree structure, it eliminates data redundancy and more readable. Furthermore, it is designed as generic as possible and can be used for any data retrieval as long as a tree structure can be defined among the tables or nodes. It can be used in a wide range of systems for database publishing, content management, supply chain management (CRM), electronic data interchange (EDI), and other e-business applications and middleware.

Claims

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A system for constructing a tree structure representing a complex query and executing the said query to retrieve data from a relational Database Management System (RDBMS) and format the retrieved data in the said tree structure, comprising:

a subprocess for defining such a query in a tree structure through configuration files;

a subprocess for defining such a query in a tree structure through a graphical user interface;

a subprocess for executing the defined query to retrieve data from a database;

a subprocess for formatting the retrieved data in the defined tree structure;

a subprocess for formatting the retrieved data in the defined tree structure and generating documents in Extensible Markup Language (XML), comprising:

a Document Type Definition (DTD) document based on the defined tree structure;

an instance of the said document type containing the retrieved data;

a subprocess for formatting the retrieved data in the defined tree structure and generating a document in HyperText Markup Language (HTML) format.

2. Computer readable code that defines the internal structure of said query in tree structure according to claim 1.

3. Internal format of the tree structure for representing a query according to claim 2.

4. The layout and pattern of the configurable properties in the configuration files for constructing a query in said tree structure according to claim 3.

5. Computer readable code for visually composing a query in tree structure according to claim 3.

6. Computer readable code for executing a query in said tree structure according to claim 3 and retrieving data from a database according to claim 1.

7. Computer readable code for formatting retrieved data according to claim 6 into a tree structure according to claim 3.

8. Computer readable code for formatting retrieved data according to claim 6 and generating XML documents, comprising the Document Type Definition declarations and an instance of said document type containing said retrieved data.

9. Computer readable code for formatting retrieved data according to claim 6 and generating an HTML document containing said retrieved data.

10. Visual presentation as shown in the accompanying drawings of a query in said tree structure according to claim 3 and claim 5.

Claims

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A system for constructing a tree structure representing a complex query and executing the said query to retrieve data from a relational Database Management System (RDBMS) and format the retrieved data in the said tree structure, comprising:

a subprocess for defining such a query in a tree structure through configuration files;

a subprocess for defining such a query in a tree structure through a graphical user interface;

a subprocess for executing the defined query to retrieve data from a database;

a subprocess for formatting the retrieved data in the defined tree structure;

a subprocess for formatting the retrieved data in the defined tree structure and generating documents in Extensible Markup Language (XML), comprising:

a Document Type Definition (DTD) document based on the defined tree structure;

an instance of the said document type containing the retrieved data;

a subprocess for formatting the retrieved data in the defined tree structure and generating a document in HyperText Markup Language (HTML) format.

2. Computer readable code that defines the internal structure of said query in tree structure according to claim 1.

3. Internal format of the tree structure for representing a query according to claim 2.

4. The layout and pattern of the configurable properties in the configuration files for constructing a query in said tree structure according to claim 3.

5. Computer readable code for visually composing a query in tree structure according to claim 3.

6. Computer readable code for executing a query in said tree structure according to claim 3 and retrieving data from a database according to claim 1.

7. Computer readable code for formatting retrieved data according to claim 6 into a tree structure according to claim 3.

8. Computer readable code for formatting retrieved data according to claim 6 and generating XML documents, comprising the Document Type Definition declarations and an instance of said document type containing said retrieved data.

9. Computer readable code for formatting retrieved data according to claim 6 and generating an HTML document containing said retrieved data.

10. Visual presentation as shown in the accompanying drawings of a query in said tree structure according to claim 3 and claim 5.

Description

BACKGROUND OF THE INVENTION

1. Field of Invention

The present invention relates generally to data retrieval and presentation in a computer system. More specifically, the present invention relates to a technique, system, and computer program for retrieval of data from a Relational Database Management System (RDBMS) and presentation of the retrieved data in tree structure in Extensible Markup Language (XML) and HyperText Markup Language (HTML) formats.

2. Prior Art

It is a common practice that personal and corporate data is stored in relational databases. These relational database management systems (RDBMS) manage data and the relationships of the data in different ways, although they usually conform to the international standard, Structured Query Language (SQL) to certain level. This makes the retrieval of complex data a demanding and often time-consuming task, depending on how the database is structured. Technologies and inventions related to query optimization have improved the performance of data retrieval, but they have limitations especially when very complex queries are involved. For example, a supplier supplies several types of products, each type with its unique set of characteristics and other related information such as their uses in different cases. If the product data is stored in a highly normalized form, data retrieval and processing for dynamically generating online catalogue can result in long-running queries and creation of large amount of temporary data in computer memory. This could significantly affect the normal business operation. To avoid this, customized system would have to be used, or the catalogue is generated on a scheduled basis at after-hour time.

For today's e-business applications such as supply chain management systems and online shopping, it is essential that data is accessed in real-time and data interchange is carried out in a way as efficient as possible. The Extensible Markup Language (XML) is an emerging technology for electronic data publishing and interchange. Data structured in XML can be used by content management application, websites, or be communicated to business partners. In order to implement

XML technologies in their environment, corporations have to build their own customized applications specific for their database.

In real world, business processes are often complex and so is the data generated and used by these processes. Some systems or applications have been developed to automate the data retrieval and formatting but they are limited to deal with simple data. Thus far, there are no efficient methods or systems generic enough to allow corporations to do complex data retrieval and format the data into XML and HTML documents on the fly to meet their business requirements. Too many customized computer code has been written and too much duplicate works have been carried out among most corporations.

Therefore, a need exists for a technique or a system by which complex data retrieval and formatting can be automated without the need of or with very little customization. Furthermore, a need exists for a system by which an ordinary database user can define such complex queries without the need of writing sophisticated SQL query statements. This invention provides a technique and system to address both issues.

SUMMARY OF THE INVENTION

An object of this invention is to provide a technique by which complex queries can be defined and executed in a flexible and efficient manner. An ease-to-use interface is provided for ordinary database users to define a complex query in a tree structure without the need of writing a complex query statement using the SQL language.

The technique disclosed in the present invention allows a high degree of flexibility in defining the relationships between a parent and its different children, which can be nested to n-depth levels. The relationships are mapped to a tree structure and the query processor executes the query based on the tree in an efficient way. The output data is also constructed in the defined tree structure in XML by default, which eliminates data redundancy and is more readable. The output data can be formatted either in XML or HTML format. Combined with XML, retrieved data can be easily transformed to other formats or databases without losing their structure and relationships.

The present invention also provides two mechanisms to allow user to define the query: through configuration files, and through a graphical user interface. The configuration mechanism through files makes it easy to implement the technology to batch processing and interaction with other applications.

Such an extremely flexible and complex query can be defined visually using the graphical user interface in the proposed system of this invention (see Fig. 1-3). The tree structure displays the nodes and their relationship and the associated properties for each node are displayed on a separate pane (Fig. 1). This enables user to edit the properties very easily. Because the system is written in the Java programming language and the Java Database Connectivity (JDBC) technologies, therefore, it can run on any platform where Java is supported and access any database it can connect using JDBC or Open Database Connectivity (ODBC) protocols, both of which are widely supported by almost all database vendors.

The technique of complex data retrieval disclosed in this invention is different from existing techniques in its high degree of flexibility and complexity in terms of the query structure, yet easy to compose, efficient processing and precise output result. Furthermore, it is generic and can be used for any data retrieval as long as a tree structure can be defined among the tables or nodes. And finally, it is written in Java thus it can be used on any platform.

The technique can be broadly used in any business systems where data is accessed from a database. These include systems for dynamic content management, supply chain management (CRM), electronic data interchange (EDI), e-commerce websites, database publishing (such as electronic and printed catalogues), database middleware, and query utility programs.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a sample display of the tree structure of a complex query and the properties for one of the nodes.

Fig. 2 shows a sample screen of columns selection.

Fig. 3 shows a sample screen of key pair selection.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

For the top-level table or root node, one or more child tables or nodes can be defined based on one or more common columns, which are typically primary and foreign keys but not limited to the key columns. Subsequently each child node can have zero to many child nodes of its own, and so on, so forth, to an unlimited levels in theory.

In order to precisely process complex queries defined by the method disclosed in this invention, the root table or node must have at least one column selected for the output. If no columns are selected, all columns are automatically included by default. The root table also must have a primary key or a column with unique values no matter whether it is included in the output or not. This is essential for the accurate generation of the tree structure and data retrieval. Special conditions, that is, the WHERE clause, can be defined for it; and the data can be ordered in a way the user wants.

For each node or table below the root level, zero or more columns can be included in the output; special conditions, that is, the WHERE clause, can be defined for it; and the data can be ordered in a way the user wants. To correctly define the relationship between a parent and a child, pairs of primary and foreign key columns must be specified. It is also acceptable to use other common key columns in both tables even though they are not directly linking the parent and the child tables. For each relationship, more than one pair of key columns can be used for accurate retrieval of desired data. When a node (except the root node) is specified with no column included in the output, this type of empty node practically acts as a linkage table.

It should be noted that the technique disclosed here defines a node in the query tree using a physical table. It can, however, be expanded to include the views as well. Since each logical view is often based on one or more physical tables or other views, it could be less efficient compared with that using the original physical tables. On the other hand, views can include aggregated columns or fields, thus are useful if aggregated values are required in the output.

The columns of a node included in the data tree structure are selected from physical tables, and optionally from views if views are also used. Aggregated columns cannot be defined directly using

the graphical tool proposed here unless they are defined in a view. It is, however, possible to define aggregated columns in configuration files.

For easier integration with other applications, the query processor is designed independently from the query definition modules. It is a set of ready-to-use components written in the Java programming language. It can be used with any application on a client computer system, which can be connected to a database server through JDBC or ODBC protocols, or integrated into server-side middleware systems. The output can be formatted either in XML or HTML format, and stored in an electronic file or redirected to other data streams, for example, to a servlet input stream.

For XML format, a valid Document Type Definition (DTD) is automatically generated based on the query definition. The generated XML document is fully compliant with the XML 1.0 standard, that is, it is both well-formed and valid. Other configurable properties are also provided.

Finally, the tree structure described here is based on a single top-level node. It can, however, be easily expanded to include multiple top-level nodes for batch processing or other cases where such a need exists, for example, for database integration, data conversion and data interchange with business partners.

SAMPLE SCREENS OF THE QUERY BUILDER INTERFACE

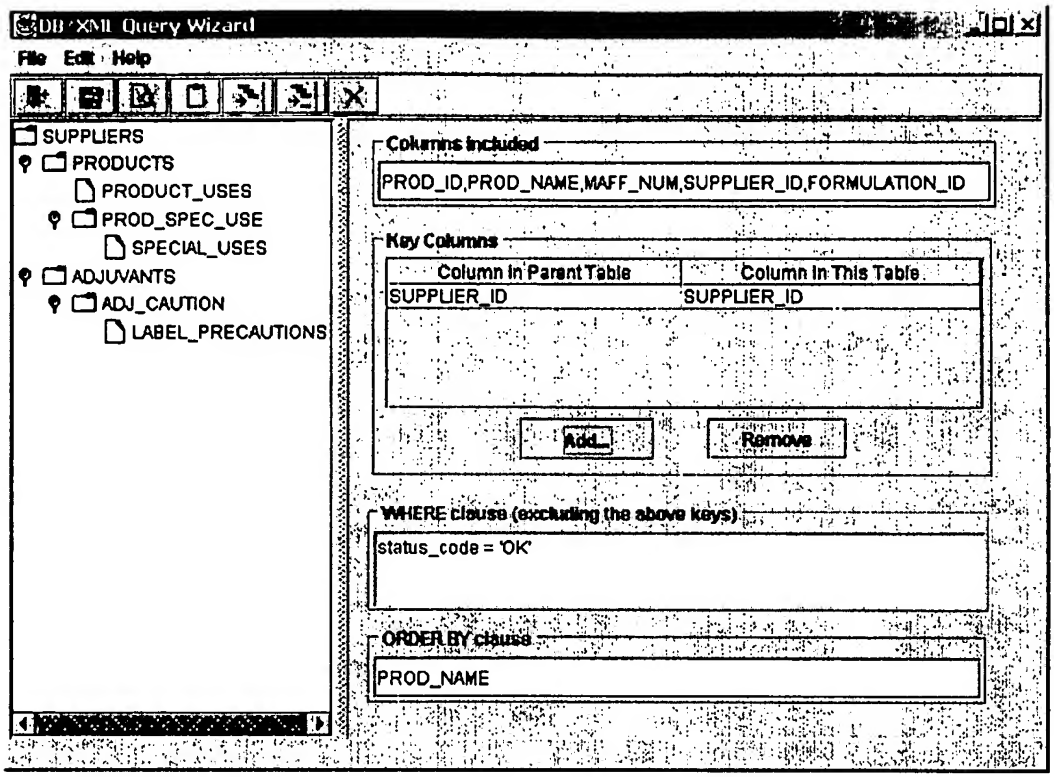


Fig. 1. Sample of tree structure of a complex query and the properties for one of the nodes

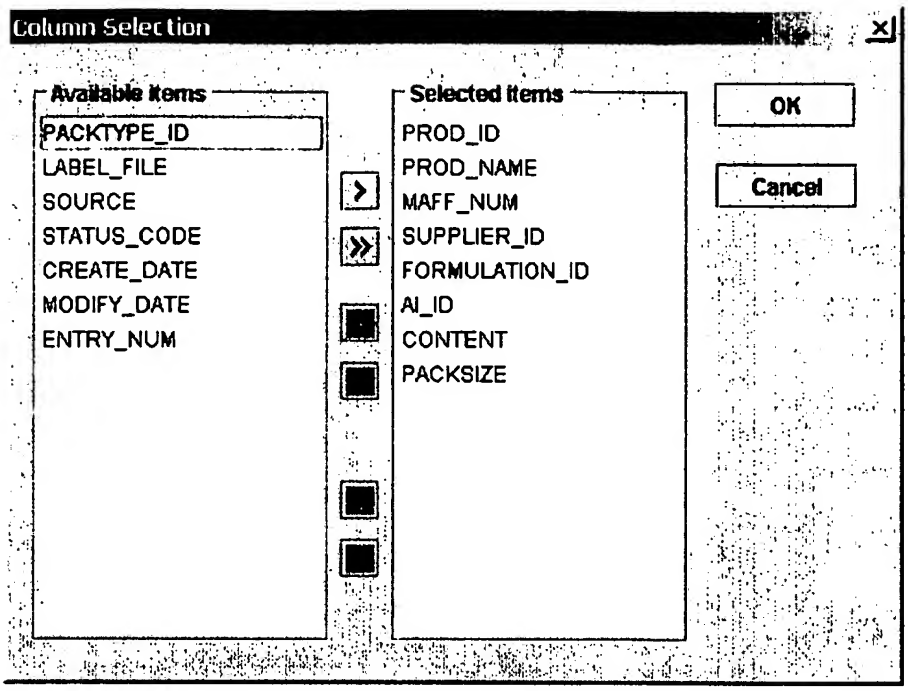


Fig. 2. Snapshot of the Table Column Selection screen

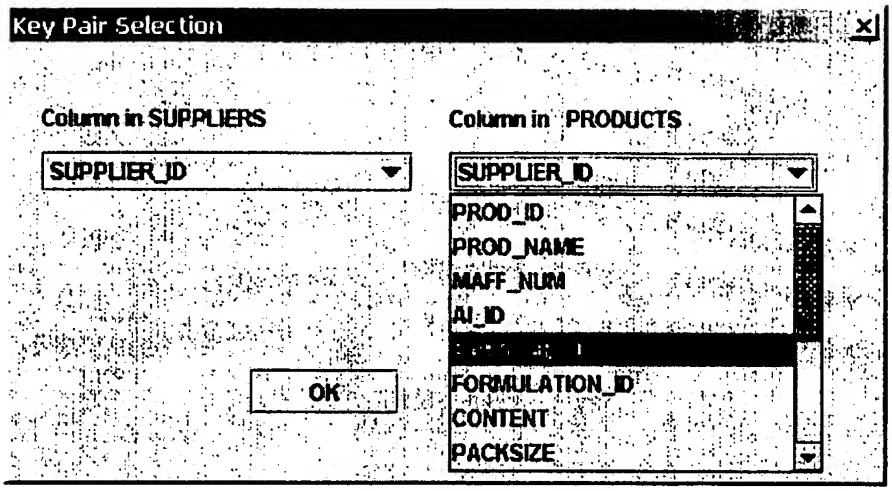
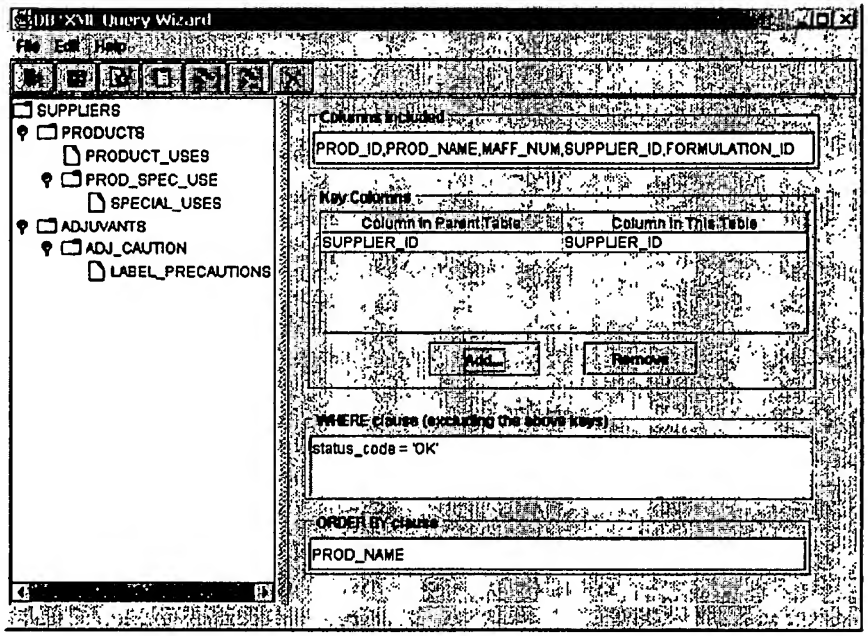


Fig. 3. Snapshot of the Key Pair Selection screen



Sample of tree structure of a complex query and the properties for one of the nodes